|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №2  по дисциплине «Языки программирования»  Вариант 2 | | |
|  | | Работу выполнил  студент группы ДВБ7-ПР3-2023 НБ 2 курса  Румянцев В.И.  «28» октября 2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1.2. 4](#_Toc181098530)

[Текст задания 4](#_Toc181098531)

[Алгоритм решения 4](#_Toc181098532)

[Тестирование 4](#_Toc181098533)

[Код программы 5](#_Toc181098534)

[Задание 1.3. 6](#_Toc181098535)

[Текст задания 6](#_Toc181098536)

[Алгоритм решения 6](#_Toc181098537)

[Тестирование 7](#_Toc181098538)

[Код программы 7](#_Toc181098539)

[Задание 2.2. 8](#_Toc181098540)

[Текст задания 8](#_Toc181098541)

[Алгоритм решения 8](#_Toc181098542)

[Тестирование 8](#_Toc181098543)

[Код программы 8](#_Toc181098544)

[Задание 3.3. 9](#_Toc181098545)

[Текст задания 9](#_Toc181098546)

[Алгоритм решения 9](#_Toc181098547)

[Тестирование 10](#_Toc181098548)

[Код программы 10](#_Toc181098549)

[Задание 4.8. 11](#_Toc181098550)

[Текст задания 11](#_Toc181098551)

[Алгоритм решения 11](#_Toc181098552)

[Тестирование 11](#_Toc181098553)

[Код программы 11](#_Toc181098554)

[Задание 5.5. 12](#_Toc181098555)

[Текст задания 12](#_Toc181098556)

[Алгоритм решения 12](#_Toc181098557)

[Тестирование 13](#_Toc181098558)

[Код программы 13](#_Toc181098559)

# Задание 1.2.

## Текст задания

Человек.

Создайте сущность Человек, которая описывается:

• Имя: строка

• Рост: целое число

Может возвращать текстовое представление вида “Name, рост: height”, где Name и height это переменная с именем и ростом.

Необходимо создать и вывести на экран следующих людей:

• Человек с именем “Клеопатра” и ростом 152

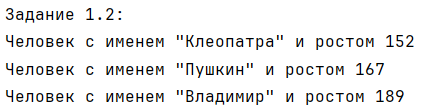
• Человек с именем “Пушкин ” и ростом 167

• Человек с именем “Владимир ” и ростом 189

## Алгоритм решения

1. Создание класса **Human** с полями **Name** и **height**.
2. Реализация конструктора для инициализации полей **Name** и **height**.
3. Реализация метода **toString** для возврата строкового представления объекта.
4. Реализация метода **printInformation** для вывода информации о человеке.
5. Создание объектов **Human** с именами "Клеопатра", "Пушкин" и "Владимир" и соответствующими ростами.
6. Вывод информации о созданных объектах на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2

# Задание 1.3.

## Текст задания

Имена.

Создайте сущность Имя, которая описывается тремя параметрами: Фамилия, Личное имя, Отчество. Имя может быть приведено к строковому виду, включающему традиционное представление всех трех параметров: Фамилия Имя Отчество (например “Иванов Иван Иванович”). Необходимо предусмотреть возможность того, что какой-либо из параметров может быть не задан, и в этом случае он не учитывается при приведении к текстовому виду.

Необходимо создать следующие имена:

• Клеопатра

• Пушкин Александр Сергеевич

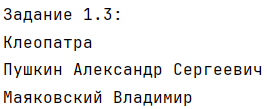
• Маяковский Владимир

Обратите внимание, что при выводе на экран, не заданные параметры никак не участвуют в образовании строки.

## Алгоритм решения

1. Создание класса **Name** с полями **LastName**, **FirstName** и **Patronymic**.
2. Реализация конструкторов для инициализации полей с различными комбинациями параметров.
3. Реализация метода **toString** для возврата строкового представления объекта, учитывая возможность отсутствия некоторых параметров.
4. Создание объектов **Name** с именами "Клеопатра", "Пушкин Александр Сергеевич" и "Маяковский Владимир".
5. Вывод информации о созданных объектах на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2

# Задание 2.2.

## Текст задания

Человек с именем.

Объедините сущности Человек из задачи 1.2 и Имя из задачи 1.3 таким образом, чтобы имя человека задавалось с использованием сущности 1.3, а не строки.

Необходимо объединить ранее созданные объекты имен и людей, с получением:

• Человека с Именем Клеопатра и ростом 152

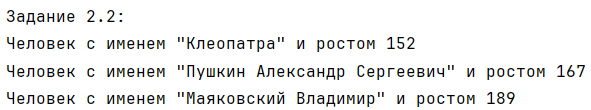
• Человека с Именем Пушкин Александр Сергеевичи ростом 167

• Человека с Именем Маяковский Владимир и ростом 189

## Алгоритм решения

1. Обновление класса **Human** для использования объекта **Name** вместо строки для имени.
2. Создание объектов **Name** с именами "Клеопатра", "Пушкин Александр Сергеевич" и "Маяковский Владимир".
3. Обновление объектов **Human** с использованием созданных объектов **Name**.
4. Вывод информации о созданных объектах на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2

# Задание 3.3.

## Текст задания

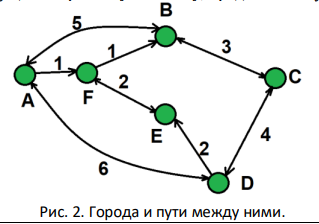
Города.

Создайте сущность Город, которая будет представлять собой точку на карте со следующими характеристиками:

• Название города

• Набор путей к следующим городам, где путь представляет собой сочетание Города и стоимости поездки в него.

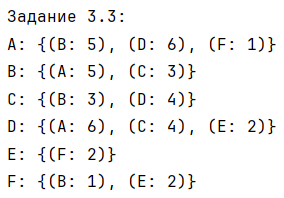
Кроме того, Город может возвращать текстовое представление, в виде названия города и списка связанных с ним городов (в виде пары: “название:стоимость”). Используя разработанную сущность реализуйте схему, представленную на рисунке 2.



## Алгоритм решения

1. Создание класса **City** с полями **name** и **paths**.
2. Создание класса **Path** с полями **destination** и **cost**.
3. Реализация метода **toString** в классе **City** для возврата строкового представления объекта, включающего название города и список связанных городов с их стоимостями.
4. Создание объектов **City** с названиями "A", "B", "C", "D", "E" и "F".
5. Создание объектов **Path** для связи городов.
6. Установка путей для каждого города.
7. Вывод информации о созданных объектах на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2

# Задание 4.8.

## Текст задания

Создаем Города.

Измените сущность Город из задачи 3.3. Новые требования включают:

• Город можно создать указав только название

• Город можно создать указав название и набор связанных с ним городов и стоимостей путей к ним

## Алгоритм решения

1. Обновление класса **City** для возможности создания объекта только с названием или с названием и набором путей.
2. Создание объекта **City** с названием "G" и набором путей.
3. Вывод информации о созданном объекте на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2

# Задание 5.5.

## Текст задания

Дроби.

Создайте сущность Дробь со следующими особенностями:

• Имеет числитель: целое число

• Имеет знаменатель: целое число

• Дробь может быть создана с указанием числителя и знаменателя

• Может вернуть строковое представление вида “числитель/знаменатель”

• Может выполнять операции сложения, вычитания, умножения и деления с другой Дробью или целым числом. Результатом операции должна быть новая Дробь (таким образом,обе исходные дроби не изменяются)

Затем необходимо выполнить следующие задачи:

1. Создать несколько экземпляров дробей.

2. Написать по одному примеру использования каждого метода.

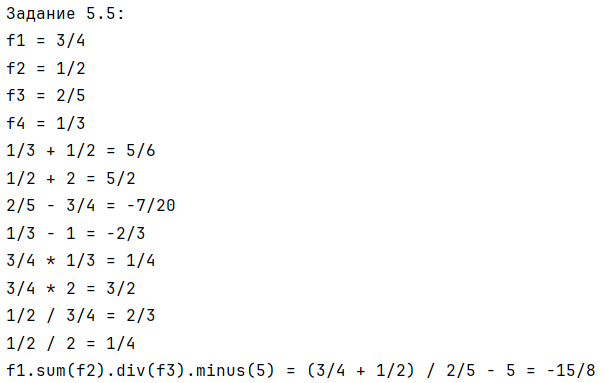
3. Вывести на экран примеры и результаты их выполнения в формате «1/3 \* 2/3 = 2/9»

4. Посчитать f1.sum(f2).div(f3).minus(5)

## Алгоритм решения

1. Создание класса **Fraction** с полями **numerator** и **denominator**.
2. Реализация конструктора для инициализации полей **numerator** и **denominator**.
3. Реализация метода **toString**для возврата строкового представления объекта.
4. Реализация методов для сложения, вычитания, умножения и деления дробей и целых чисел.
5. Реализация метода **simplify** для сокращения дроби.
6. Создание объектов **Fraction** с различными числителями и знаменателями.
7. Выполнение операций сложения, вычитания, умножения и деления с созданными объектами.
8. Вывод результатов операций на экран.
9. Выполнение сложной операции **f1.sum(f2).div(f3).minus(5)** и вывод результата на экран.

## Тестирование



## Код программы

https://github.com/Vladimir-Rumyantsev/JavaLabs/tree/master/src/LabWork2